

Manual de Usuario OpenPKIXML_Client (Ecuador)



1.	Introducción de producto OpenPKIXML Client Applet	3
1.1.	Generalidades	3
1.2.	Configuración de producto.....	3
1.3.	Servicio de soporte y Algoritmo	3
1.4.	Requisitos del sistema	6
2.	Instalación de OpenPKIXML_Client	7
2.1.	Instalación	7
3.	Programas ejemplos de OpenPKIXML_Client.....	13
3.1.	Ejemplo de creación/validación de firma digitalXML	13
3.1.1.	Ejemplo de creación/validación de firma digital en forma Enveloped.....	13
3.1.2.	Ejemplo de creación/validación de firma digital en forma Enveloping.....	15
3.1.3.	Ejemplo de creación/validación de firma digital XPath.....	17
3.2.	Ejemplo de Cifrado/Descifrado de XML	19
3.2.1.	Ejemplo de cifrado/descifrado de elementos (Uso de certificado de servidor)..	19
3.2.2.	Ejemplo de cifrado/descifrado de contenidos (Uso de certificado de servidor).	20
3.2.3.	Ejemplo de cifrado/descifrado de elementos (Uso de certificado personal)..	21
3.2.4.	Ejemplo de cifrado/descifrado de contenidos (Uso de certificado personal).	22

1. Introducción de producto OpenPKIXML Client Applet

1.1. Generalidades

- 1.1.1 Ofrece el producto CoOpenPKIXML Client como Client de seguridad de XML con el fin de Login basado en certificado digital, creación/validación de firma de datos XML, cifrado/descifrado (encriptación/desencriptación).
- 1.1.2 Ofrece los siguientes por cada operación:
- Instalación de Applet de Client basado en Web : `sgj_safexml_install.js`
 - Ejecución de ventanilla para elegir certificado digital : `deployJava.js`
 - Control de ventanilla para elegir certificado digital : `sgj_safexml_cert.js`
 - Login de certificado digital : `sgj_safexml_login.js`
 - Firma de XML y validación : `sgj_safexml_sign.js`
 - Cifrado/Descifrado (encriptación/desencriptación) : `sgj_safexml_encrypt.js`
- 1.1.3 Ofrece `sgj_safexml_util.js` para el trámite de leer archivo XML.

1.2. Configuración de producto

OpenPKIXML Client distribuido consiste en los siguientes:

Directorio(archive)	Descripción
script	Se incluye archives js para usar Applet.
lib	Librerías a utilizar en Applet (Necesita explicación por jar)
sample	Se incluye páginas ejemplos html para realizar pruebas con Applet.
Server	Se incluye página jsp para que se valide datos creado en Client dentro el modulo de Servidor.
Manual_Usuario_OpenPKIXML_Client_v1.1_Ecuador.doc	El manual de usuario
OpenPKIXML_Client Class Reference	Documento referencial a HTML Class

1.3. Servicio de soporte y Algoritmo

- 1.3.1 OpenPKIXML_Client proporciona el estándar de firma XML y cifrado “XML Signature Syntax and Processing <http://www.w3.org/TR/xmlsig-core/>” y servicio de login con certificado digital “XML Encryption Syntax and Processing <http://www.w3.org/TR/xmlenc-core/>”. En siguiente listado se detallan opciones de soporte de OpenPKIXML_Client:

Opción	Descripción	Nota
Ventana de selección de certificado	Opción para ventana de selección de certificado digital.	sgj_safexml_cert.js,
Firma XML y validación	Opción básica para firmar y validar XML.	sgj_safexml_sign.js
Cifrado/Descifrado de XML.	Opción básica para cifrado/descifrado de XML.	sgj_safexml_encr ypt.js

1.3.2 En siguiente listado se encuentra algoritmos de firma que soporta OpenPKIXML_Client:

Algoritmo	URI	Nota
RSA-SHA1	http://www.w3.org/2000/09/xmldsig#rsa-sha1	sgj_safexml_algorithm.js
RSA-SHA256	http://www.w3.org/2001/04/xmldsig-more#rsa-sha256	sgj_safexml_algorithm.js
KCDSA-SHA1	http://www.tta.or.kr/2002/11/xmldsig#kcdsa-sha1	sgj_safexml_algorithm.js
Normalización	http://www.w3.org/TR/2001/REC-xml-c14n-20010315	sgj_safexml_algorithm.js
Normalización	http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments	sgj_safexml_algorithm.js

1.3.3 En siguiente listado se encuentra algoritmos de cifrado que soporta OpenPKIXML_Applet.

Algoritmo	URI	Nota
DES	No está definido.	sgj_safexml_algorithm.js
SEED	No está definido.	sgj_safexml_algorithm.js
ARIA128	No está definido.	sgj_safexml_algorithm.js
ARIA192	No está definido.	sgj_safexml_algorithm.js
ARIA256	No está definido.	sgj_safexml_algorithm.js

AES128	http://www.w3.org/2001/04/xmlenc#aes128-cbc	sgj_safexml_algorithm.js
AES192	http://www.w3.org/2001/04/xmlenc#aes192-cbc	sgj_safexml_algorithm.js
AES256	http://www.w3.org/2001/04/xmlenc#aes256-cbc	sgj_safexml_algorithm.js

1.3.4 En siguiente listado se encuentra algoritmos de Message Digest que soporta OpenPKIXML_Client.

Algoritmo	URI	Nota
SHA1	http://www.w3.org/2000/09/xmldsig#sha1	sgj_safexml_algorithm.js
SHA256	http://www.w3.org/2000/09/xmlenc#sha256	sgj_safexml_algorithm.js
SHA384	http://www.w3.org/2000/09/xmldsig-more#sha384	sgj_safexml_algorithm.js
SHA512	http://www.w3.org/2000/09/xmlenc#sha512	sgj_safexml_algorithm.js

1.4. Requisitos del sistema

El sistema operativo ofrecido por Módulo OpenPKIXML_Client es lo siguiente:

División	Requisitos	
Sistema Operativo(OS)	- Windows XP o superior - Linux : ubuntu 10 o superior - Mac OS : Leopard o superior	
Hardware	- RAM : mínimo 512MB o superior - CPU : Pentium 1Ghz o superior	
Web Browser	Windows	IE6 o superior Safari 5.0 o superior Firefox 5.0 o superior Chrome 9.0 o superior Opera 10.0 o superior
	Linux	Firefox 3.5 o superior
	Mac	Safari 5.0 o superior Firefox 3.5 o superior Chrome 9.0 o superior
JRE	Java Runtime Environment 1.6.X JVM Soporte de 32bit o 64bit JVM	
Open Port	- Out-Bound LDAP(389) Port - Dirección de CRL LDAP de ACs acreditadas	

[Cuadro 1] Requisitos

2. Instalación de OpenPKIXML_Client

2.1. Instalación

2.1.1 Descomprimir OpenPKIXML.tar ofrecido en ROOT HOME de WAS correspondiente.

Comando : tar -xvf OpenPKIXML.tar

Generar el directorio OpenPKIXML luego de descomprimirlo.

```
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT$ ls -al
? (4) 16940
drwxr-xr-x 15 sg sg 4096 2011-11-14 15:06 .
drwxr-xr-x 8 sg sg 4096 2011-02-08 22:14 ..
-rw-r--r-- 1 sg sg 55517 2011-07-29 18:38 Copy of CreditTransfer.htm
-rw-r--r-- 1 sg sg 55411 2011-07-29 18:38 CreditTransfer.htm
drwxr-xr-x 2 sg sg 4096 2011-07-29 18:38 META-INF
drwxr-xr-x 3 sg sg 4096 2011-11-14 15:06 OpenPKIXML
-rw-r--r-- 1 sg sg 10747520 2011-11-14 15:06 OpenPKIXML.tar
drwxr-xr-x 6 sg sg 4096 2011-10-05 19:55 WEB-INF
drwxr-xr-x 5 sg sg 4096 2011-11-11 14:45 applettest
-rw-r--r-- 1 sg sg 7554 2011-07-29 18:38 authNumber.htm
-rw-r--r-- 1 sg sg 10528 2011-08-01 14:54 authNumber.jsp
-rw-r--r-- 1 sg sg 11062 2011-08-01 14:54 creditTransferwait.jsp
-rw-r--r-- 1 sg sg 166707 2011-07-29 18:38 creditTransfer_org.htm
drwxr-xr-x 2 sg sg 4096 2011-08-02 15:23 css
drwxr-xr-x 3 sg sg 4096 2011-08-26 16:43 file
drwxr-xr-x 2 sg sg 4096 2011-07-29 18:38 image
-rw-r--r-- 1 sg sg 75281 2011-08-01 14:56 index.html
-rw-r--r-- 1 sg sg 31569 2011-08-24 20:02 jeexplorer.jsp
drwxr-xr-x 2 sg sg 4096 2011-11-14 15:04 jsp
-rw-r--r-- 1 sg sg 37068 2011-07-29 18:38 logout.htm
-rw-r--r-- 1 sg sg 7514 2011-07-29 18:38 mkey.jsp
drwxr-xr-x 3 sg sg 4096 2011-09-22 13:37 pds
-rw-r--r-- 1 sg sg 8111 2011-07-29 18:38 popup.htm
-rw-r--r-- 1 sg sg 432 2011-07-29 18:38 progress.htm
-rw-r--r-- 1 sg sg 2862 2011-09-16 11:23 ptest.html
-rw-r--r-- 1 sg sg 32016 2011-07-29 18:38 regcert.htm
drwxr-xr-x 5 sg sg 4096 2011-07-29 18:38 resource
drwxr-xr-x 6 sg sg 4096 2011-07-29 18:42 servlet
drwxr-xr-x 2 sg sg 4096 2011-07-29 18:38 sh
-rw-r--r-- 1 sg sg 1896 2011-07-29 18:38 sign.htm
drwxr-xr-x 5 sg sg 4096 2011-10-07 17:45 smartsign
-rw-r--r-- 1 sg sg 9 2011-07-29 18:38 test.jsp
-rw-r--r-- 1 sg sg 1527 2011-08-22 17:41 xmlhttp.html
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT$
```

2.1.2 Entrar al directorio OpenPKIXML y verificar si se encuentra directorio Sample.

```
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT/openPKIXML$ ls -al
? (4) 12
drwxr-xr-x 3 sg sg 4096 2011-11-14 15:06 .
drwxr-xr-x 15 sg sg 4096 2011-11-14 15:06 ..
drwxr-xr-x 5 sg sg 4096 2011-11-14 15:06 Sample
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT/openPKIXML$
```

2.1.3 Dentro del directorio Sample se encuentra los siguientes archivos y directorios.

```
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT/OpenPKIXML/sample$ ls -al
? (4) 56
drwxr-xr-x 5 sg sg 4096 2011-11-14 15:06 .
drwxr-xr-x 3 sg sg 4096 2011-11-14 15:06 ..
-rwxr-xr-x 1 sg sg 1345 2011-11-14 15:05 Jarcheck2.jsp
drwxr-xr-x 3 sg sg 4096 2011-11-14 15:06 Nation
drwxr-xr-x 2 sg sg 4096 2011-11-14 15:06 Server
-rwxr-xr-x 1 sg sg 478 2011-10-04 15:56 index.html
-rwxr-xr-x 1 sg sg 1273 2011-11-14 10:16 install.html
-rwxr-xr-x 1 sg sg 1442 2011-11-12 16:31 left.html
-rwxr-xr-x 1 sg sg 2643 2010-12-06 13:41 portscan.jsp
-rwxr-xr-x 1 sg sg 7936 2011-09-27 10:33 propertyinfo.jsp
-rwxr-xr-x 1 sg sg 2683 2010-01-18 14:40 providerinfo.jsp
-rwxr-xr-x 1 sg sg 2602 2011-11-12 16:39 right.html
drwxr-xr-x 3 sg sg 4096 2011-11-14 15:06 script
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT/OpenPKIXML/sample$
```

- 2.1.4 En el directorio Script se encuentra archivos js sobre opciones a utilizar en Client incluso archivos de licencia. El archivo de licencia es licensecode.js y copia license.js emitido por KICA (Korea Information Certificate Authority) en este lugar.

(Ofrece license.js como licencia temporal para pruebas o licencia perpetua.)

```
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT/OpenPKIXML/sample/script$ ls -al
? (4) 508
drwxr-xr-x 3 sg sg 4096 2011-11-14 15:06 .
drwxr-xr-x 5 sg sg 4096 2011-11-14 15:06 ..
-rwxr-xr-x 1 sg sg 17787 2011-11-14 11:22 LicenseDecoder.js
-rwxr-xr-x 1 sg sg 4190 2011-11-14 11:22 LicenseValidator.js
-rwxr-xr-x 1 sg sg 27895 2011-08-24 09:10 deployjava.js
-rwxr-xr-x 1 sg sg 272554 2011-11-14 11:23 kica_min.js
-rwxr-xr-x 2 sg sg 4096 2011-11-14 15:06 lib
-rwxr-xr-x 1 sg sg 6838 2011-11-14 11:22 license.js
-rwxr-xr-x 1 sg sg 2148 2011-11-14 11:23 licenseinterface.js
-rwxr-xr-x 1 sg sg 2536 2011-11-14 11:36 licensecode.js
-rwxr-xr-x 1 sg sg 6413 2011-09-26 13:35 sgj_safexml_algorithm.js
-rwxr-xr-x 1 sg sg 17449 2011-11-14 11:23 sgj_safexml_cert.js
-rwxr-xr-x 1 sg sg 22601 2011-11-14 11:22 sgj_safexml_encrypt.js
-rwxr-xr-x 1 sg sg 10150 2011-11-09 16:02 sgj_safexml_install.js
-rwxr-xr-x 1 sg sg 14837 2011-09-26 20:19 sgj_safexml_pkcs7.js
-rwxr-xr-x 1 sg sg 61745 2011-11-14 11:22 sgj_safexml_sign.js
-rwxr-xr-x 1 sg sg 11038 2011-11-14 11:22 sgj_safexml_util.js
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT/OpenPKIXML/sample/script$
```

2.1.5 En el directorio lib se encuentra archivos jar a utilizar en Client, los cuales serán distribuidos y instalados en Client.

```
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT/OpenPKIXML/Sample/script/lib$ ls -al
?4) 3664
drwxr-xr-x 2 sg sg 4096 2011-11-14 15:06 .
drwxr-xr-x 3 sg sg 4096 2011-11-14 15:06 ..
-rwxr-xr-x 1 sg sg 4721 2011-01-17 15:19 codesign
-rwxr-xr-x 1 sg sg 1110350 2011-11-14 00:30 ewcommon.jar
-rwxr-xr-x 1 sg sg 1183663 2011-11-03 20:04 images.jar
-rwxr-xr-x 1 sg sg 153797 2011-11-14 00:41 launcher.jar.pack.gz
-rwxr-xr-x 1 sg sg 106305 2011-11-14 00:41 ldapjdk.jar.pack.gz
-rwxr-xr-x 1 sg sg 32605 2011-11-14 00:42 libgpkapi_jni.jar.pack.gz
-rwxr-xr-x 1 sg sg 854 2011-10-17 19:27 packjar.bat
-rwxr-xr-x 1 sg sg 523134 2011-11-14 00:30 sgapplet.jar
-rwxr-xr-x 1 sg sg 12563 2011-11-14 00:42 sgppki.jar.pack.gz
-rwxr-xr-x 1 sg sg 133644 2011-05-31 16:32 singgatecrypto.jar.pack.gz
-rwxr-xr-x 1 sg sg 104 2011-10-17 19:20 signjar.bat
-rwxr-xr-x 1 sg sg 13904 2011-11-13 22:09 xmlsecurity_applet_archive-1.0.jar.pack.gz
-rwxr-xr-x 1 sg sg 51374 2011-11-13 23:37 xmlsecurity_applet_file-1.0.jar
-rwxr-xr-x 1 sg sg 385892 2011-11-14 00:30 xmlsecurity_client_api-1.0.jar
sg@sg-laptop:/usr/local/tomcat-5.5.33/webapps/ROOT/OpenPKIXML/Sample/script/lib$
```

2.1.6 Cambiar ruta del archivo sgj_safexml_install.js en el directorio Script.

- Codebase : Se designa ruta relativa una vez que se encuentre Client Library a ser distribuido en ./script/lib.
- ProductHome : Se designa ruta absoluta una vez que se encuentre Client Library a ser distribuido en ./script/lib.

```

if(mode == "normal")
{
    attributes =
    {
        codebase: './script/lib',
        id: object_id,
        code: 'com.sg.xml.security.applets.LauncherApplet',
        archive: 'xmlsecurity_applet_archive-1.0.jar,launcher.jar,ldapjdk.jar,libgpkiapi_
        rypto.jar',
        width:0, height:0
    };

    parameters =
    {
        ProductHome: './OpenPKIXML/Sample/script/lib/',
        sgAppletFile: 'sgapplet.jar/images.jar/ewscommon.jar;xmlsecurity_applet_file-1.0.
        1.0.jar',
        sgAppletFileVersion: '1.0.48;1.0.48;1.0.48;1.0.0;1.0.0',
        nativeLib: 'BHSML_JNI2.dll-2.0.0.1,KICAUAC.dll-1.0.0.3,KicaUACJNI.dll-1.0.0.2,gpk
        .dll-1.1.2.0,nsldap32v11.dll-4224.0.0.8803',
        java_arguments: '-Djnlp.packEnabled=true',
        separate_jvm: 'true',
        MAYSCRIPT: 'yes',
        codebase_lookup: 'false',
        scriptable: 'true'
    };
}
else
{
    attributes =
    {
        codebase: './script/lib',
        id: object_id,
        code: 'com.sg.xml.security.applets.LauncherApplet',
        //archive: 'xmlsecurity_applet_archive_beta.jar,signgateCrypto_v1.2.jar,launcher.j
        ni.jar,sggpki.jar,sgsecukit_v3.92.jar',
        archive: 'xmlsecurity_applet_archive-1.0.jar,launcher.jar,ldapjdk.jar,libgpkiapi_
        rypto.jar',
        id: object_id,
        width:0, height:0
    };

    parameters =
    {
        ProductHome: './OpenPKIXML/Sample/script/lib/',
        //sgAppletFile: 'sgapplet.jar/images.jar;xmlsecurity_applet_file_beta.jar;ewscomm
        sgkm.jar;signgate_common.jar;sgExCrypto.jar',
        //sgAppletFileVersion: '1.0.48;1.0.48;1.0.48;1.0.48;1.0.48;1.0.48;1.0.48;1.0.48',
        sgAppletFile: 'sgapplet.jar/images.jar/ewscommon.jar;xmlsecurity_applet_file-1.0.

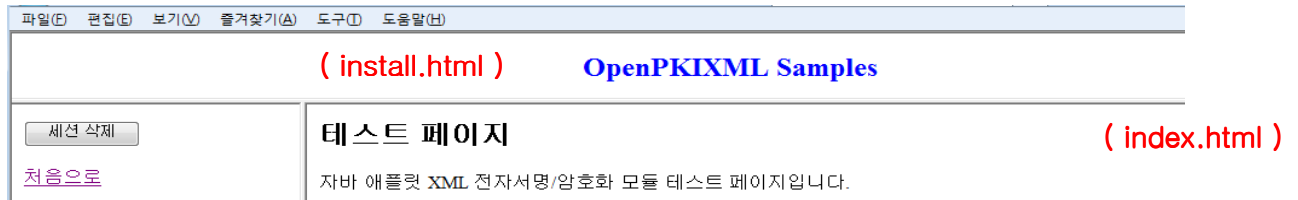
```

[3. Pantalla de designar ruta de Client Library a ser distribuido]

2.1.7 Modificar página a prestar servicio. (Referencia a página ejemplo a ofrecer)

Si supone en tal forma de configuración de página a prestar servicios,

- **Frame de todas las páginas : index.html**
- **Frame definido por página : install.html**



- **Agregar siguiente código javascript a install.html.**

```
<script type="text/javascript" src="./script/deployJava.js"></script>
<script type="text/javascript" src="./script/sgj_safexml_install.js"></script>
<script type="text/javascript" src="./script/sgj_safexml_cert.js"></script>
<script type="text/javascript" src="./script/sgj_safexml_algorithm.js"></script>
<script type="text/javascript" src="./script/sgj_safexml_sign.js"></script>
<script type="text/javascript" src="./script/sgj_safexml_encrypt.js"></script>
<script type="text/javascript" src="./script/sgj_safexml_pkcs7.js"></script>
<script type="text/javascript" src="./script/sgj_safexml_util.js"></script>
<script type="text/javascript" src="./script/kica_min.js"></script>
<script type="text/javascript" src="./script/license.js"></script>
<script type="text/javascript" src="./script/licensecode.js"></script>
<script type="text/javascript" src="./script/LicenseDecoder.js"></script>
<script type="text/javascript" src="./script/licenseInterface.js"></script>
<script type="text/javascript" src="./script/LicenseValidator.js"></script>
```

- **Otorgar atributo de nombre(NAME) sobre install.html a Index.html.**

```
</HEAD>
<FRAMESET ROWS="10%,90%">
<FRAME SRC="install.html" NAME="applet">
<FRAMESET cols="20%,*">
<FRAME SRC="left.html" NAME="left_pages">
<FRAME SRC="right.html" NAME="right_pages">
</FRAMESET>
</FRAMESET>
```

- Puede utilizar con la llamada de objeto en siguiente forma con el uso de valor de nombre otorgado al install.html en la página real.

```

<script language="javascript">
var SGObj = parent.frames["applet"];
function SGJ_XMLSecurity()
{
    var encoding = document.fill_in.Encoding.value;
    var encType = document.fill_in.EncType.value;
    var encPosition = document.fill_in.EncPosition.value;

```

- La configuración mencionada(2.1.6 ~ 2.1.7) están dispuestas en la página ejemplo ofrecida.



3. Programas ejemplos de OpenPKIXML_Client

- **Facilitará a comprenderse el programa ejemplo con referencia a páginas muestras ofrecidas.**

3.1. Ejemplo de creación/validación de firma digitalXML

3.1.1. Ejemplo de creación/validación de firma digital en forma Enveloped

Es el ejemplo de creación/validación de firma digital en forma Enveloped con datos de SOAP y datos generales XML.

Casos de prueba

- Creación/Validación de firma digital de forma básica Enveloped
- Creación/Validación de firma digital basada en algoritmo de Signature/Digest
- Creación/Validación de firma digital basada en Sign Position
- Creación/Validación de firma digital basada en Reference URI.

Código muestra(Sample Code)

1. Creación de firma digital

```
// Abrir ventana de selección de certificado digital.
var selected = SGOBJ.SGJ_SelectCertDlg("sg");
if(selected == false) return;

// Generar objeto de firma digital
// Debe configurar encoding definido. Si no existe, se configura UTF-8.
var ret = SGOBJ.SGJ_CreateSignatureObj(encoding);
if(ret == false) return;

// Configuración de información necesaria para la firma digital
var ret = SGOBJ.SGJ_SetStringSignature(xmlData, signPosition,
    SGOBJ.ALGORITHM_SIGINFO_ENVELOPED, signAlgo, isAppendSibling);
if(ret == false) return;

// Configuración de información referencial y Algoritmo de Message Digest.
```

```
var reference = SGOBJ.SGJ_CreateXMLReference(referURI, digestAlgo);
```

```
if(reference == null) return;
```

```
// Setear Transform necesario para creación de firma XML.
```

```
ret = SGOBJ.SGJ_SetXMLTransform(reference, null, null, null);
```

```
if(ret == false) return;
```

```
// Trámite de creación de firma digital
```

```
// Setear si incluye certificado de firma digital según isIncludeCert (true or false)
```

```
var signedData = SGOBJ.SGJ_Sign(isIncludeCert);
```

```
if(signedData == null){
```

```
    alert("valor de resultado de firma es nulo.");
```

```
    return;
```

```
}
```

2. Validación de firma digital

```
// Generar el objeto de validación de firma digital.
```

```
// Debe configurar encoding definido. Si no existe, se configura UTF-8.
```

```
ret = SGOBJ.SGJ_CreateVerifyObj(encoding);
```

```
if(ret == false) return;
```

```
// Configuración de datos necesarios de firma digital para la validación
```

```
ret = SGOBJ.SGJ_SetStringVerify(signedData);
```

```
if(ret == false) return;
```

```
// Trámite de validación de firma digital
```

```
var result = SGOBJ.SGJ_Verify();
```

```
if(result == true) alert("Éxito en validación de firma digital XML");
```

```
else alert("Fallo en validación de firma digital XML");
```

3.1.2. Ejemplo de creación/validación de firma digital en forma Enveloping

Es el ejemplo de creación/validación de firma digital en forma Enveloping con datos de SOAP y datos generales XML.

Casos de prueba

- Creación/Validación de firma digital de forma básica Enveloping.
- Creación/Validación de firma digital basada en algoritmo de Signature/Digest.
- Creación/Validación de firma digital basada en Enveloping Id.

Código muestra(Sample Code)

1. Creación de firma digital

```
// Abrir ventana de selección de certificado digital.
var selected = SGOBJ.SGJ_SelectCertDlg("sg");
if(selected == false) return;

// Generar objeto de firma digital
// Debe configurar encoding definido. Si no existe, se configura UTF-8.
var ret = SGOBJ.SGJ_CreateSignatureObj(encoding);
if(ret == false) return;

// Configuración de información necesaria para la firma digital
var ret = SGOBJ.SGJ_SetStringSignature("", "", SGOBJ.ALGORITHM_SIGINFO_ENVELOPING,
signAlgo, isAppendSibling);
if(ret == false) return;

// Configuración de Enveloping Id y datos para firma digital
ret = SGOBJ.SGJ_SetEnvelopingStringData(envelopingID, xmlData);
if(ret == false) return;

// Configuración de información referencial y Algoritmo de Message Digest.
var reference = SGOBJ.SGJ_SetXMLReference(referURI, digestAlgo);
```

```
if(reference == null) return;
```

```
// Setear Transform necesario para creación de firma XML.
```

```
ret = SGOBJ.SGJ_SetXMLEnvelopingTransform(reference);
```

```
if(ret == false) return;
```

```
// Trámite de creación de firma digital
```

```
Setear si incluye certificado de firma digital según isIncludeCert (true or false)
```

```
var signedData = SGOBJ.SGJ_Sign(isIncludeCert);
```

```
if(signedData == null){
```

```
    alert("valor de resultado de firma es nulo.");
```

```
    return;
```

```
}
```

2. Validación de firma digital

```
// Generar el objeto de validación de firma digital.
```

```
// Debe configurar encoding definido. Si no existe, se configura UTF-8.
```

```
ret = SGOBJ.SGJ_CreateVerifyObj(encoding);
```

```
if(ret == false) return;
```

```
// Configuración de datos necesarios de firma digital para la validación
```

```
ret = SGOBJ.SGJ_SetStringVerify(signedData);
```

```
if(ret == false) return;
```

```
// Trámite de validación de firma digital
```

```
var result = SGOBJ.SGJ_Verify();
```

```
if(result == true) alert("Éxito en validación de firma digital XML");
```

```
else alert("Fallo en validación de firma digital XML");
```

3.1.3. Ejemplo de creación/validación de firma digital XPath

Es el ejemplo de creación/validación de firma digital en forma de configuración Xpath que designa el alcance de firma de SOAP y los datos generales XML.

Casos de prueba

- Creación/Validación de firma digital para configuración de alcance de Xpath.
- Creación/Validación de firma digital basada en algoritmo de Signature/Digest.
- Creación/Validación de firma digital basada en Sign Position.
- Creación/Validación de firma digital basada en Reference URI.
- Creación/Validación de firma digital basada en configuración de datos de XPath.

Código muestra (Sample Code)

1. Creación de firma digital

```
// Abrir ventana de selección de certificado digital.
var selected = SGOBJ.SGJ_SelectCertDlg("sg");
if(selected == false) return;

// Generar objeto de firma digital
// Debe configurar encoding definido. Si no existe, se configura UTF-8.
var ret = SGOBJ.SGJ_CreateSignatureObj(encoding);
if(ret == false) return;

// Configuración de información necesaria para la firma digital
var ret = SGOBJ.SGJ_SetStringSignature(xmlData, signPosition, "", signAlgo, isAppendSibling);
if(ret == false) return;

// Configuración de información referencial y Algoritmo de Message Digest.
var reference = SGOBJ.SGJ_SetXMLReference(referURI, digestAlgo);
if(reference == null) return;

// Setear Transform necesario para creación de firma XML.
// Incluso información de XPath.
```

```
ret = SGOBJ.SGJ_SetXMLTransform(reference, xpath, xpath_ns, xpath_ns_uri);
```

```
if(ret == false) return;
```

```
// Trámite de creación de firma digital
```

```
// Setear si incluye certificado de firma digital según isIncludeCert (true or false)
```

```
var signedData = SGOBJ.SGJ_Sign(isIncludeCert);
```

```
if(signedData == null){
```

```
    alert("valor de resultado de firma es nulo.");
```

```
    return;
```

```
}
```

2. Validación de firma digital

```
// Generar el objeto de validación de firma digital.
```

```
// Debe configurar encoding definido. Si no existe, se configura UTF-8.
```

```
ret = SGOBJ.SGJ_CreateVerifyObj(encoding);
```

```
if(ret == false) return;
```

```
// Configuración de datos necesarios de firma digital para la validación
```

```
ret = SGOBJ.SGJ_SetStringVerify(signedData);
```

```
if(ret == false) return;
```

```
// Trámite de validación de firma digital
```

```
var result = SGOBJ.SGJ_Verify();
```

```
if(result == true) alert("Éxito en validación de firma digital XML");
```

```
else alert("Fallo en validación de firma digital XML");
```

3.2. Ejemplo de Cifrado/Descifrado de XML

3.2.1. Ejemplo de cifrado/descifrado de elementos (Uso de certificado de servidor)

Es el ejemplo de configurar la zona de cifrado de SOAP y datos generales XML y realizar el cifrado/descifrado de elementos en la zona configurada.

Casos de prueba

- Cifrado/Descifrado de todas las sintaxis del documento con uso del certificado del servidor.
- Cifrado/Descifrado de elementos según la configuración de zona de cifrado.

Código muestra (Sample Code)

1. Cifrado de XML (Uso de certificado del servidor)

```
// Generar objeto de cifrado de XML
var ret = SGOBJ.SGJ_CreateEncryptionObj(encoding);
if(ret == false) return;

// Configuración de datos de cifrado de XML, Algoritmo de cifrado y Algoritmo de normalización
ret = SGOBJ.SGJ_SetStringEncryption(xmlData, SGOBJ.ENCRYPTION_ALGO_URI_AES256,
SGOBJ.ALGORITHM_CANONICAL_C14N);
if(ret == false) return;

// Configuración de forma y localidad de cifrado de XML
ret = SGOBJ.SGJ_SetEncryptType(SGOBJ.ENCRYPTION_TYPE_ELEMENT, encPosition);
if(ret == false) return;

// Configuración de información de clave de certificado recibido desde servidor
ret = SGOBJ.SGJ_SetEncKeyInfo(kmCertPEM, certClass);
if(ret == false) return;

// Trámite de cifrado de XML
var encData = SGOBJ.SGJ_Encrypt();
if(encData == null) return;
```

	KICA-CUM-RG-20101015	Rev.	V1.0
	Manual de usuario (OpenPKIXML Client para Ecuador)		

3.2.2. Ejemplo de cifrado/descifrado de contenidos (Uso de certificado de servidor)

Es el ejemplo de configurar la zona de cifrado de SOAP y datos generales XML y realizar el cifrado/descifrado de contenidos en la zona configurada.

Casos de prueba

- Cifrado/Descifrado de todas las sintaxis del documento con uso de certificado del servidor.
- Cifrado/Descifrado de contenidos según la configuración de zona de cifrado

Código muestra (Sample Code)

1. Cifrado de XML (Uso de certificado del servidor)

```
// Generar objeto de cifrado de XML
var ret = SGOBJ.SGJ_CreateEncryptionObj(encoding);
if(ret == false) return;

// Configuración de datos de cifrado de XML, Algoritmo de cifrado y Algoritmo de normalización
ret = SGOBJ.SGJ_SetStringEncryption(xmlData, SGOBJ.ENCRYPTION_ALGO_URI_AES256,
SGOBJ.ALGORITHM_CANONICAL_C14N);
if(ret == false) return;

// Configuración de forma y localidad de cifrado de XML
ret = SGOBJ.SGJ_SetEncryptType(SGOBJ.ENCRYPTION_TYPE_CONTENT, encPosition);
if(ret == false) return;

// Configuración de información de clave de certificado recibido desde servidor
ret = SGOBJ.SGJ_SetEncKeyInfo(kmCertPEM, certClass);
if(ret == false) return;

// Trámite de cifrado de XML
var encData = SGOBJ.SGJ_Encrypt();
if(encData == null) return;
```

	KICA-CUM-RG-20101015	Rev.	V1.0
	Manual de usuario (OpenPKIXML Client para Ecuador)		

3.2.3. Ejemplo de cifrado/descifrado de elementos (Uso de certificado personal)

Es el ejemplo de configurar la zona de cifrado de SOAP y datos generales XML y realizar el cifrado/descifrado de elementos en la zona configurada.

Casos de prueba

- Cifrado/Descifrado de todas las sintaxis del documento con uso del certificado del servidor.
- Cifrado/Descifrado de elementos según la configuración de zona de cifrado.

Código muestra (Sample Code)

1. Cifrado de XML (Uso de certificado del servidor)

```
// Generar objeto de cifrado de XML
var ret = SGOBJ.SGJ_CreateEncryptionObj(encoding);
if(ret == false) return;

// Configuración de datos de cifrado de XML, Algoritmo de cifrado y Algoritmo de normalización
ret = SGOBJ.SGJ_SetStringEncryption(xmlData, SGOBJ.ENCRYPTION_ALGO_URI_AES256,
SGOBJ.ALGORITHM_CANONICAL_C14N);
if(ret == false) return;

// Configuración de forma y localidad de cifrado de XML
ret = SGOBJ.SGJ_SetEncryptType(SGOBJ.ENCRYPTION_TYPE_ELEMENT, encPosition);
if(ret == false) return;

// Configuración de información de clave de certificado recibido desde servidor
ret = SGOBJ.SGJ_SetEncKeyInfo(null, null);
if(ret == false) return;

// Trámite de cifrado de XML
var encData = SGOBJ.SGJ_Encrypt();
if(encData == null) return;
```

	KICA-CUM-RG-20101015	Rev.	V1.0
	Manual de usuario (OpenPKIXML Client para Ecuador)		

3.2.4. Ejemplo de cifrado/descifrado de contenidos (Uso de certificado personal)

Es el ejemplo de configurar la zona de cifrado de SOAP y datos generales XML y realizar el cifrado/descifrado de elementos en la zona configurada.

Casos de prueba

- Cifrado/Descifrado de todas las sintaxis del documento con uso del certificado del servidor.
- Cifrado/Descifrado de contenidos según la configuración de zona de cifrado.

Código muestra (Sample Code)

1. Cifrado de XML (Uso de certificado del servidor)

```
// Generación de objeto de cifrado de XML
var ret = SGOBJ.SGJ_CreateEncryptionObj(encoding);
if(ret == false) return;

// Configuración de datos de cifrado de XML, Algoritmo de cifrado y Algoritmo de normalización
ret = SGOBJ.SGJ_SetStringEncryption(xmlData, SGOBJ.ENCRYPTION_ALGO_URI_AES256,
SGOBJ.ALGORITHM_CANONICAL_C14N);
if(ret == false) return;

// Configuración de forma y localidad de cifrado de XML
ret = SGOBJ.SGJ_SetEncryptType(SGOBJ.ENCRYPTION_TYPE_CONTENT, encPosition);
if(ret == false) return;

// Configuración de información de clave de certificado recibido desde servidor
ret = SGOBJ.SGJ_SetEncKeyInfo(null, null);
if(ret == false) return;

// Trámite de cifrado de XML
var encData = SGOBJ.SGJ_Encrypt();
if(encData == null) return;
```

	KICA-CUM-RG-20101015	Rev.	V1.0
	Manual de usuario (OpenPKIXML Client para Ecuador)		

Este Manual de usuario OpenPKIXML_Client fue elaborado en octubre del 2010 y KICA S.A. cuenta con toda la propiedad intelectual del Manual. Dicha empresa tiene derecho de modificar contenidos del Manual sin aviso con el fin de mejorar el rendimiento de producto. La copia sin permiso, reproducción y publicación de partes o todo el documento sin permiso de KICA S.A. se viola el derecho de propiedad intelectual.



Dirección : KICA S.A. Piso #16. Business Tower Nurikkum Square,
1605 Sangam-dong, Mapo-gu, Seoul, Korea
Tel. +82) 2-360-3000
Fax. +82) 2-360-3209
Página web : www.signgate.com